Yukti – The Chip Design Challenge 2025

Round 1: 4-bit Arithmetic Logic Unit (ALU)

Team Name: SILICON SAMURAI
College: SNU Chennai
Tool Used: Xilinx Vivado 2025.1
Date: 3rd November 2025

1. Objective

To design and simulate a 4-bit modular Arithmetic Logic Unit (ALU) using Verilog HDL. The ALU performs arithmetic and logical operations controlled by a 3-bit opcode signal.

2. Design Overview

The ALU accepts two 4-bit operands (A and B) and a 3-bit opcode. Each operation (Addition, Subtraction, Multiplication, Division, AND, OR, XOR, Rotate Left) is implemented in a separate Verilog module for modularity and clarity. The top-level module integrates all submodules to perform operations based on the opcode.

3. Opcode Mapping

Opcode	Operation	Description
000	ADD	Adds A and B
001	SUB	Subtracts B from A
010	MUL	Multiplies A and B
011	DIV	Divides A by B
100	AND	Bitwise AND
101	OR	Bitwise OR
110	XOR	Bitwise XOR
111	ROL	Rotate Left (Custom Operation)

4. Simulation and Verification

The design was simulated in Vivado using a structured testbench. Inputs (A, B, opcode) were varied every 10 ns to verify all 8 operations. Outputs matched expected results, confirming correct ALU behavior.

Figure 1: Behavioral simulation waveform of the 4-bit ALU.

5. Design Assumptions

- Operands A and B are 4-bit inputs.
- Opcode is 3-bit, providing 8 unique operations.
- Output is 8-bit to accommodate larger results like multiplication.
- Division by zero outputs 0.

- Overflow and carry are not explicitly handled.

6. Key Learnings

• Learned modular and hierarchical Verilog design.
• Understood testbench creation and behavioral simulation.
• Gained hands-on experience with Vivado waveform analysis.
• Improved debugging and verification skills for HDL projects.

7. Future Improvements

- Add overflow and carry flag handling.
- Extend ALU to 8-bit or 16-bit width.
- Include pipeline stages for better performance.